Cross-Layer Optimization of Radio Sleep Intervals to Increase Thin Client Energy Efficiency

Pieter Simoens, Farhan Azmat Ali, Bert Vankeirsbilck, Lien Deboosere, Filip De Turck, *Member, IEEE*, Bart Dhoedt, *Member, IEEE*, Piet Demeester, *Fellow, IEEE,* and Rodolfo Torrea-Duran

***Abstract*—Thin client computing trades local processing for network bandwidth consumption by offloading application logic to remote servers. User input and display updates are exchanged between client and server through a thin client protocol. This thin client protocol traffic can lead to a significantly higher power consumption of the radio interface of the wireless device. In this contribution, we present a cross-layer algorithm that exploits thin client protocol layer information to determine intervals where no traffic from the server is expected. During these intervals, the wireless network interface card (WNIC) is instructed to enter the energy conserving sleep mode. Using this algorithm for a remote text editor, WNIC energy consumption reductions of 21-52 % can be achieved.**

***Index Terms*—Thin client, cross-layer, energy saving.**

I. Introduction

Thin client computing refers to the paradigm in which the user device relies on a remote server to perform a significant fraction of its computational tasks. Mostly, the device functionality is limited to the transmission of user input to the application server, and the presentation of the display updates that are calculated by the server.

Especially in the mobile context, the thin client concept is very promising. Users are able to access demanding applications from mobile devices, which often lack the required processing resources to execute the application locally. Because only basic functionality and processing power is required at the terminal, thin client devices can be made lightweight and potentially energy efficient.

When the conventional approach is followed, i.e. applications are run locally, the communication between the client and the network is typically limited to storage or retrieval requests. On the other hand, a thin client must be connected to the network at all times to communicate with the application server. This more intensive network communication might result in a significant increase of the energy consumption due to the wireless network interface card (WNIC). Although the exact share in the total power budget depends on the client hardware and the volume of data exchanged, it is generally acknowledged that the WNIC is a significant source of consumed energy on mobile devices [1]. For example, when streaming video to a Compaq iPaq, the WNIC amounts for 37.7% of the total power consumption [2]. In typical smartphone usage, the WiFi interface accounts for 25% of the total power budget [3]. On an Asus EEE PC, a device without hard disk, merely switching on the WiFi interface without actually sending or transmitting data, already increases the total device power consumption by 12% [4]. As a consequence, WNIC energy optimization strategies can contribute to the overall power efficiency of thin clients.

In previous work [5], we have demonstrated that even in scenarios with complex display updates, still half of the total WNIC energy consumption is due to the time spent in idle mode. Consequently, major additional energy savings are to be expected from approaches that put the WNIC in the energy-conserving sleep mode during intervals without communication, on top of existing algorithms that optimize encoding and transmission parameters. In this paper, we assume that all applications are offloaded to an application server. Only a single network connection is required to communicate user events and display updates over a thin client protocol. We present a cross-layer algorithm that analyzes the thin client protocol information and puts the WNIC in sleep mode when no traffic from the server is expected. In Section II, we provide details on the thin client protocol operation. The cross-layer algorithm is presented in Section III and is validated in Section IV for a remotely running office application.

II. Thin Client Protocol Operation

Thin client protocols can operate either in push or pull mode. In push protocols, such as Microsoft's Remote Desktop Protocol, the server autonomously determines the rate at which display updates are sent to the client. In pull protocols, such as Virtual Network Computing (VNC) [6], the client explicitly requests a new display update from the server.

In this paper, we will focus on VNC, which is a widely used and open source pull protocol that uses TCP as transport layer protocol. Each time the VNC client receives an update from the server, it immediately requests the next update. Figure 1 depicts the two ways a VNC server can respond when it receives such a request.

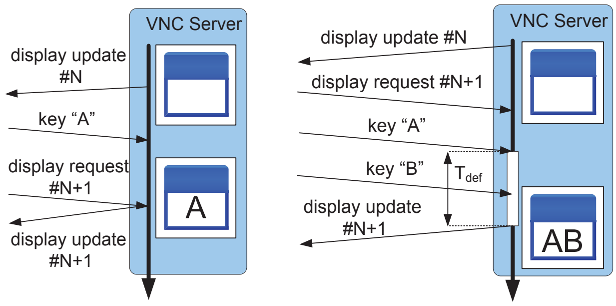
If the display has changed since the previous update was sent, the request will be answered immediately. If the display is unmodified, a *deferred update* mechanism is activated to avoid superfluous updates. In this case, a timer is started when the display is updated for the first time after receiving the request. This timer expires after a period of T_{def}, after which a display update is sent. This update includes all display changes that occurred during the timer interval.

Manuscript received August 10, 2010. The associate editor coordinating the review of this letter and approving it for publication was F. Granelli.

P. Simoens, F. A. Ali, B. Vankeirsbilck, L. Deboosere, F. De Turck, B. Dhoedt, and P. Demeester are with the Dept. of Information Tech., Ghent University, Belgium (e-mail: pieter.simoens@intec.ugent.be).

R. Torrea-Duran is with IMEC, Leuven, Belgium.

Digital Object Identifier 10.1109/LCOMM.2010.11.101450



(a) Immediate display update (b) Deferred display update

Fig. 1. Immediate and deferred server responses to display update requests. In these example scenarios, the user presses the “A” and “B” keys.

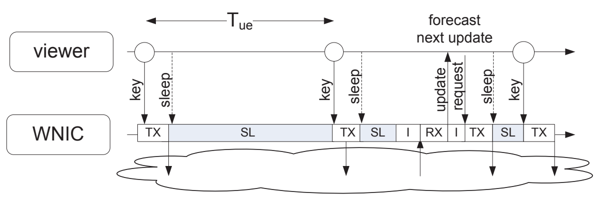


Fig. 2. The viewer forecasts the arrival time of the next update. The WNIC is put in sleep mode during intervals without communication.

III. Cross-Layer Algorithm

From the viewpoint of the client, the update following a display request must not be expected earlier than one network roundtrip time (RTT), as can be seen in Fig. 1. The RTT is composed of the latency between client and access point (RTT_{CA}), which is primarily determined by the path loss on the wireless channel, and the latency between access point and server (RTT_{AS}). During this interval, the WNIC on the client can be put in sleep mode without risking to miss any incoming data. It needs to wake up only when an update is expected, or when a transmission opportunity for user event occurs. In the following discussion, a transmission opportunity for user events is foreseen every T_{ue}. In between two transmission opportunities, user events are buffered at the thin client protocol layer.

The cross-layer approach between thin client protocol layer and MAC/PHY layer is presented in Fig. 2.

After every transmission of a user event or a display update request, the WNIC is put in sleep mode, either until the next user event transmission opportunity or until the display update will be transmitted. The algorithm itself is composed of two parts, and is presented in Fig. 3. In the first part, the arrival time of the next display update is forecasted. To this end, the VNC protocol was slightly modified by adding to each display update the sequence number of the last user event that was received by the server. If not all transmitted user events were contained in the last display update, a display update request will be answered immediately and the update is expected over RTT. In the other case, the deferred update mechanism will be triggered by the first user event that is transmitted after this request. In the second part of the algorithm, the WNIC is actually put in sleep mode for the appropriate interval.

Require: RTT and T_{def} are initialized
  $deferredUpdate =$ **false**
  $lastDisplayUserEvent = 0$
  $CT \leftarrow$ getCurrentTime()
  if data.type == display update **then**
    $nextUpdateScheduled =$ **false**
    update $lastDisplayUserEvent$
  else if data.type == display request **then**
    if $lastDisplayUserEvent < seqUserEvent$ **then**
      $timeNextUpdate \leftarrow CT + RTT$
      $nextUpdateScheduled =$ **true**
    else
      $deferredUpdate =$ **true**
      $nextUpdateScheduled =$ **false**
    end if
  else if data.type == user input **then**
    $seqUserEvent++$
    if !$nextUpdateScheduled$ AND $deferredUpdate$ **then**
      $timeNextUpdate \leftarrow CT + RTT + T_{def}$
      $nextUpdateScheduled =$ **true**
    end if
  end if
  /* Determine sleep time */
  if $nextUpdateScheduled$ **then**
    $sleepTime = min(timeNextUpdate, timeNextUE) - CT$
  else
    $sleepTime = timeNextUE - CT$
  end if

Fig. 3. XL algorithm to forecast the arrival time of the next update, invoked each time data is sent or received at the client.

IV. Simulation Results and Discussion

We have implemented the cross-layer algorithm in TightVNC 1.3.10 for Unix, a bandwidth-efficient VNC implementation, and validated on a test set-up composed of a client and server machine that are interconnected by an impairment node and a 802.11 wireless tranceiver (MAC+PHY) and channel simulator implemented in ns-2. To span a wide range of settings, the parameter RTT_{AS} was varied between 0 and 100 ms and the path loss, influencing (RTT_{CA}) between 60 and 90 dB. With $RTT_{AS} = 0$, the use case is simulated in which the server is located in the same WLAN as the thin client. T_{ue} was varied between 20 and 100 ms. Each experiment was repeated 10 times to reduce the influence of random variations. Currently, the value of RTT_{AS} is statically configured, but this could be replaced by a standard RTT monitor tool offering up to date estimates. The ns-2 simulator was equiped with energy models of a WNIC equiped with software defined radio that is optimized for energy-efficiency [7]. In the experiments, Open Office Writer, a text editor, was used. A user event was transmittted at every transmission opportunity, by generating keystrokes with a period of T_{ue}.

Figure 4 illustrates the total energy reduction when the cross-layer algorithm is activated (XL). Also, the energy consumption of the platform is indicated when only a reference MAC-PHY optimization strategy is applied [8], denoted by RS.

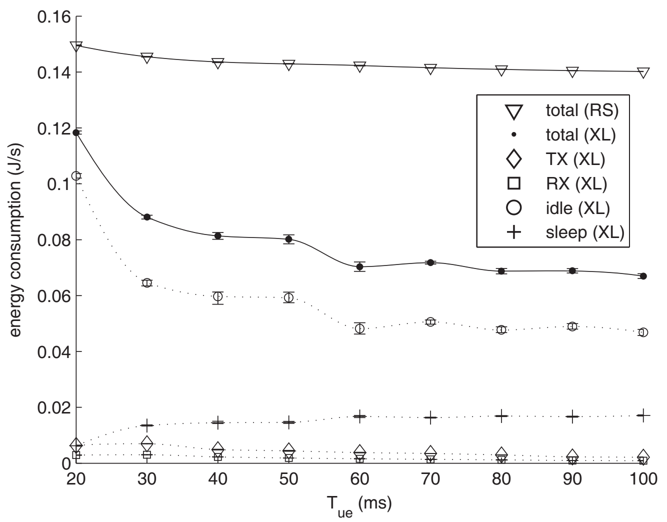


Fig. 4. Energy consumption of the wireless energy platform with RTT_{AS} = 30 ms and 60 dB path loss. Higher values of T_{ue} result in longer periods without communication to put the WNIC in sleep mode.

TABLE I
REDUCTION IN TOTAL POWER CONSUMPTION OF XL VS RS[%] ($T_{ue} = 40$ MS)

		path loss [dB]			
		60	70	80	90
RTT_{AS}	0	42.40	42.65	42.38	40.87
	30	44.88	43.79	43.72	38.90
	50	45.40	44.81	44.94	43.86
	70	36.25	36.65	36.49	35.39
	100	48.42	47.83	48.04	45.22

For XL, also the energy consumption in each of the 4 states of the wireless platform is shown. Increasing values of T_{ue} result in less data to be sent, because user data is generated less frequently. This also means that the display will change less frequently. Hence, the amount of display updates decreases accordingly, less data is received by the client and less energy is consumed by the WNIC in the receive state. Nevertheless, in the case of RS, higher values of T_{ue} are not reflected in a significant decrease of the total power consumption. This confirms the observation that an important fraction of the WNIC energy consumption is due to the time spent in the idle mode, as explained in Section I. By contrast, the XL algorithm is able to exploit these longer intervals without communication, and power savings are achieved of 21% for $T_{ue} = 20$ ms and up to 52% for $T_{ue} = 100$ ms. On Fig. 4, the decreasing trend is more pronounced for $T_{ue} = 60$ ms. In this case, the arrival of display updates coincides with the transmission of a user event. The WNIC must not wake up between two user events to receive an update, and longer sleeping times can be achieved.

Results for other values of the path loss and the network RTT are presented in Table I. The relative reduction in total power consumption varies between 35% and 49%. The energy gains for the particular case of $RTT_{AS} = 0$ ms are in line with the gains achieved for other RTT_{AS} values. This demonstrates the benefits of the deferred update mechanism of Fig. 1(b), limiting the number of request-response pairs per time unit. For $RTT_{AS} = 70$ ms, the energy gains are slightly lower, because the VNC server continuously switches between immediate and deferred updates. This complicates the forecast algorithm and decreases the energy gain. Because the results on the total energy reduction are based on proprietary energy models, we have added more hardware independent results on the WNIC idle time in Table II. With RS, the WNIC is more than 98% of the time in idle state, due to the low rate of thin client protocol traffic. XL succeeds in reducing this idle time to 37.8-54.7%. Currently, the algorithm was implemented in a conservative way to preserve the user responsiveness, which limits the achieved energy gains. If the time error between the predicted and the actual moment of receiving the next update was higher than 10 ms, the XL algorithm was disabled until two more display updates are received. If the WNIC is in sleep mode while a display update is being transmitted, this will be observed as packet loss at the TCP layer and as increased latency at the higher layer. Consequently, the TCP algorithm will apply the congestion control algorithm and reduce its send window, resulting in a lower throughput and a decreased user responsiveness for the next updates as well. By disabling the XL algorithm for the next two updates, we allow TCP to recover more quickly from the path loss and increase its send window back to the original level. This is beneficial to the user perceived responsiveness.

TABLE II
WNIC IDLE TIME [%] ($T_{ue} = 40$ MS)

		path loss [dB]							
		60		70		80		90	
		RS	XL	RS	XL	RS	XL	RS	XL
RTT_{AS}	0	98.6	44.4	98.6	44.0	98.6	44.3	98.2	45.6
	30	98.8	41.4	98.6	43.2	98.7	43.0	98.3	48.3
	50	98.7	40.4	98.7	41.8	98.7	41.5	98.3	42.3
	70	98.8	53.3	98.7	53.6	98.7	53.7	98.4	54.7
	100	99.0	37.8	99.0	39.4	98.9	38.9	98.7	42.1

ACKNOWLEDGMENT

Part of the research leading to these results was done for the MobiThin project and has received funding from the European Community's Seventh Framework (FP7/2007-2013).

REFERENCES

[1] H. Yan *et al.*, "Client-centered, energy-efficient wireless communication on IEEE 802.11b networks," *IEEE Trans. Mobile Computing*, vol. 5, no. 11, 2006.

[2] S. Mohapatra *et al.*, "Dynamo: a cross-layer framework for end-to-end QoS and energy optimization in mobile handheld devices," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, 2007.

[3] A. Anand *et al.*, "A quantitative analysis of power consumption for location-aware applications on smart phones," in *IEEE International Symposium on Industrial Electronics*, 2007.

[4] Asus, "Hardware power consumption of the asus eee pc," http://wiki.eeeuser.com/hardware_power_consumption.

[5] P. Simoens *et al.*, "Characterization of power consumption in thin clients due to protocol data transmission over IEEE 802.11," in *Intl. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless*, 2009.

[6] T. Richardson *et al.*, "Virtual network computing," *IEEE Internet Computing*, vol. 2, no. 1, 1998.

[7] V. Derudder *et al.*, "A 200 Mbps+ 2.14nj/b digital baseband multi processor system-on-chip for SDRS," in *Symp. VLSI Circuits*, 2009.

[8] A. Dejonghe *et al.*, "Green reconfigurable radio systems," *IEEE Signal Process. Mag.*, vol. 24, no. 3, 2007.

EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Euro. Trans. Telecomms. **00**: 1–13 (0000)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/ett.0000

Power Efficiency of Thin Clients

Willem Vereecken[1] Lien Deboosere[1] Pieter Simoens[1] Brecht Vermeulen[1]
Didier Colle[1] Chris Develder[1] Mario Pickavet[1] Bart Dhoedt[1] Piet Demeester[1]

[1]*Ghent University - IBBT, Department of Information Technology (INTEC)*
Gaston Crommenlaan 8, Bus 201, 9050 Ghent, Belgium
Email: {firstname.lastname}@intec.ugent.be

SUMMARY

Worldwide, awareness for energy consumption is raising because of global energy production limits as well as because of environmental concerns. As the energy fraction currently consumed by ICT related equipment is substantial (about 8% of electricity consumption worldwide in the use phase) and the growth rate in this particular sector is spectacular, also in the ICT sector adequate solutions are needed to allow sustainable growth. In this paper we aim at reducing power consumption of desktop applications by applying a thin client approach and we analyze the conditions necessary. To this end, estimates on power consumptions in typical desktop scenarios and analogous thin client settings are made and analyzed. The paper concludes with an experimental study on currently available equipment, to translate the generic conclusions into their current implications and trade-offs. Copyright © 0000 AEIT

1. Introduction

Only recently awareness is raising that power consumption directly related to ICT equipment and services represents a relevant fraction of the worldwide energy production. Given the annual growth rate of these services[1], in some cases exceeding 10% on a yearly basis, ICT related power consumption is indeed becoming an increasingly worrying concern. As more and more businesses are relying on sustained ICT services, energy concerns might constrain economical growth in a number of vital economical sectors, thereby jeopardizing the wealth to a considerable extent.

The explosive growth in ICT related energy consumption can be explained by a number of trends: not only the worldwide adoption rate of existing services (including broadband Internet services, mobile communication services, ...) is to blame, but also the emergence of new, resource and energy-hungry services [2]. Amongst the latter category, an important example is the birth of upload and consumption services of personal content (still images and video), requiring huge data centers and high speed network facilities [3]. Also replacement of existing equipment by state-of-the-art devices generally implies an increase in power consumption. Especially since the year 2000 the power consumption of PCs is rapidly rising [4].

The combination of these mechanisms has brought us in a situation where the ICT related energy consumption can be estimated at 8% of the primary electricity production in 2008. Forecasts for 2020 are typically in the range of 14% [5]. It is clear that this growth rate will be difficult to sustain, also in view of rising energy prices in combination with environmental concerns.

In this paper we will demonstrate the thin client paradigm as a solution to increase power efficiency. First we construct an analytical model comprising the home environment, the network and the data center. In sections 3 & 4 we will analyse the active state and passive state power consumption. Secondly, in section 5, we will evaluate the model based on real world measurements. In section 6 the major conclusions will be summarized.

Prepared using ettauth.cls *[Version: 2007/01/05 v1.00]*

Received September 25, 2008
Revised December 23, 2009
Accepted 2010

2. Related Work

Solutions to save power almost always use the same underlying technique: i.e. scale down the performance of devices as much as possible or even shut down equipment when possible. This technique is well known in mobile computing, arguably the sector where terminal power consumption is of prime importance (to improve battery lifetime): wireless transmission protocols switch to a lower transmission speed when possible (and go to standby mode), and the wireless card is even shut down when no network activity is detected [6]. However, these techniques are considered as well in other areas [7], [8].

A second solution to save power is the principle of virtualisation on which the thin client paradigm is based. In virtualisation the logical representation of a piece of hardware and the physical piece of hardware itself is decoupled. Thus one can share a single piece of hardware over different logical representations instead of using dedicated hardware. This results in a reduction in the number of devices and thus power saving. This technique is used in [9]. A different form of virtualization is grid computing. An example of grid computed used as a way to save power is presented in [10].

Currently most initiatives focus on isolated areas of ICT. On the network level, a good example is the IEEE study group on Energy Efficient Ethernet [11], where power savings for Ethernet are studied, again based on scaling down the link bit rate in a coordinated way. In the data center application area it is worth mentioning the Green Grid consortium [12] focussing on advancing energy efficiency in data centers.

In this paper we will investigate the thin client computing paradigm as an avenue for power saving. This approach is in fact not unlike the mainframe approach generally adopted in the ’60s-’70s (and left again in the early ’80s), where a server farm is performing the computational intensive (and hence energy hungry) functions, while the rendering for the end-user is done on very constrained devices.

A key challenge in thin client architectures, is the delay between a user event (e.g. keystroke) and the corresponding display update [13]. Ideally, the user should perceive at least the same application responsiveness as when running the application locally. This leads to requirements for time delays and QoS. In [14], it is shown that current thin client protocols were designed for low motion scenarios (e.g. office applications) and are not suited for high motion scenarios (e.g. multimedia applications). Therefore, a hybrid thin client protocol is developed in [15]. This protocol switches dynamically between a classic thin client protocol and video streaming based on the amount of motion in the application. This protocol enables execution of multimedia applications in thin client environments.

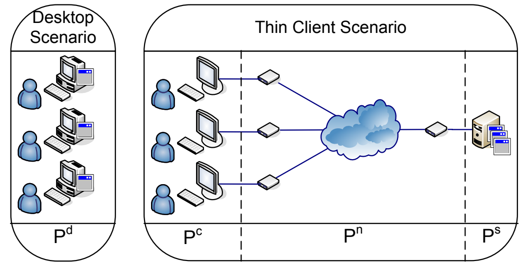


Figure 1. Desktop and Thin Client Scenario.

In [16] an extensive study of environmental impacts of thin clients is presented using life cycle assessment. In this paper, however, we will focus on the power consumption in the use phase. In [17], [18] and [19] modelling for PCs and servers is discussed. In [20] a modelling for access networks is done. An energy efficiency model for peer-to-peer networks is constructed in [21]. In the latter case virtualisation is used as well but in a different manner as in the thin client paradigm: a peer-to-peer approach is used instead of a client-server approach.

The novelty of this paper is the construction of an end-to-end analytical model for the power consumption of the thin client paradigm. Unlike the previously mentioned work we take the different premises on which power is consumed into account instead of focussing solely on a single domain such as the network or the data centre.

3. Active state analysis

When evaluating the power efficiency of the thin client paradigm we consider two scenarios. On the one hand we have a traditional desktop where each user is running a standalone application on a standard PC. In the second scenario the desktops are replaced with thin client terminals. The standalone applications run remotely on servers in the data center. Both scenarios are schematically depicted in Figure 1. In this section we consider all users to be active and we refer to this analysis as the *active state analysis*.

It is already clear that the thin client scenario has a number of advantages (+) and disadvantages (–) when compared to the desktop scenario:

+ The power consumed by the thin client terminal is significantly lower than a normal desktop PC.

DOI: 10.1002/ett

+ Server side resources can be delivered more efficiently: high-end servers are shared between all users, implying that the server infrastructure will exhibit less idle periods (the typical load on desktops is below 20%).

– Applications are run remotely, implying possibly application specific network overhead (e.g. for sending input events to the server and getting screen updates back). Additional equipment is needed (e.g. switch at server side, network interface cards consuming power ...).

– Protocol overhead from the thin client protocol requires additional server side processing.

– Resources at the server side must be cooled, increasing the power budget for the thin client scenario.

Given these observations, one can already conclude that the balance for the thin client paradigm will certainly depend on the following factors: server resource efficiency (influenced by the achievable amount of sharing and optimal resource usage), server side cooling efficiency and the bandwidth consumption (assuming power scales with consumed bandwidth, it is clear that applications requiring high bandwith thin client protocols, e.g. multimedia editing, will benefit less).

3.1. *Desktop Scenario*

When in the active state, the main sources of power consumption are the CPU, the hard disk and the network interface. (Note that power consumption caused by the monitor is not taken into account, as a similar amount of power would be consumed in the thin client monitor.) Each of these hardware elements is characterized by a load (a real number between 0 and 1), i.e. λ^d_{CPU}, λ^d_{HD} and λ^d_{NIC} denoting the load on the CPU, the hard disk and the network interface card respectively. The unloaded power consumption for each of these is written as $P^d_{0,CPU}$, $P^d_{0,HD}$ and $P^d_{0,NIC}$ and the power consumed in loaded conditions is therefore (with '*' representing CPU, HD or NIC):

$$P^d_* = P^d_{0,*} + f^d_* \left(\lambda^d_*\right) \tag{1}$$

Note that the CPU power consumption incurred by network traffic is included in P^d_{NIC} . The functions f^d_* simply express the relation between device load and power consumption.

The overall power consumption of single desktop therefore equals

$$P^d = \sum_{*=CPU,HD,NIC} \left[P^d_{0,*} + f^d_* \left(\lambda^d_*\right)\right] \tag{2}$$

$$= P^d_0 + \sum_{*=CPU,HD,NIC} f^d_* \left(\lambda^d_*\right) \tag{3}$$

with P^d_0 as the total unloaded power consumption of the desktop:

$$P^d_0 = P^d_{0,CPU} + P^d_{0,HD} + P^d_{0,NIC} \tag{4}$$

The power consumed by the hard disk is heavily dominated by the rotation motor of the drive, and far less by the load. Therefore f^d_{HD} can be omitted.

For the rest we assume a linear model for the power consumption as is suggested in [18]. In [17] it is argumented that the linear model has an acceptable margin of error and more complex models only improve this margin slightly. In this paper we denote the coefficient of the relevant parameters as α^*_*.

Since we assume a standalone application, the network card is unloaded. The desktop power consumption becomes:

$$P^d = P^d_0 + \alpha^d_{CPU}\lambda^d_{CPU} \tag{5}$$

3.2. *Thin Client Scenario*

In the thin client scenario several types of equipment need to be considered. Firstly we will consider the client terminal and the server. These will behave like a desktop PC. However we also need to consider the power consumption caused by the load λ^*_{NIC} on the network interface card (NIC). Secondly the power consumption of the network needs to be modelled. Thirdly we need to take into account that certain equipment types are located in a data center. This equipment is being cooled and the power consumption of the cooling also needs to be incorporated in the model.

3.2.1. *The Client Terminal*

A thin client terminal typically behaves like a desktop PC without a hard drive. After experimental measurements the influence of the CPU load f^c_{CPU} and NIC load f^c_{NIC} appears to be neglectable. Therefore we assume the power consumption to be constant:

$$P^c = P^c_0 \tag{6}$$

Euro. Trans. Telecomms. **00**: 1–13 (0000)
DOI: 10.1002/ett

3.2.2. The server

For the servers we again assume a linear model. [17] arguments that for multicore servers the margin of error becomes larger. However, since we are using the server for multiple instances of a desktop PC the programs running on the server will have a similar complexity and the threads running on the different cores can be considered independent. Therefore, in this particular case, the margin of error of the model will be comparable to the desktop PC case. For the NIC we assume a linear model as is done in [19]. The power consumption becomes:

$$P^s = P_0^s + \alpha_{CPU}^s \lambda_{CPU}^s + \alpha_{NIC}^s \lambda_{NIC}^s \tag{7}$$

The load λ_{NIC}^s is in reality the bandwidth received by the server b^s. We express this bandwidth as a function of the bandwidth b perceived at the client. When assuming a share ratio of N users per server we get:

$$\lambda_{NIC}^s = b^s = Nb \tag{8}$$

Obviously, the load on the server λ_{CPU}^s is related to the load on the clients. The amount of work to be performed by a single server is at least the amount of work done by N desktops. On the other hand, there is the extra work needed on the server to support N sessions, and processing the protocol overhead (to receive input from the thin clients and to construct and send back screen updates). If we note the processing capacity of a server (according to a relevant performance-oriented benchmark such as SPECint2000 [22]) as C^s and the analogous parameter for the desktop case C^d we have

$$\lambda_{CPU}^s C^s > N \lambda_{CPU}^d C^d \tag{9}$$

By denoting the extra load caused per user by ϵ, we have

$$\lambda_{CPU}^s = N \left[\lambda_{CPU}^d \frac{C^d}{C^s} + \epsilon \right] \tag{10}$$

Since $\lambda_{CPU}^s \leq 1$ we get a maximal value for N:

$$N \leq \left[\lambda_{CPU}^d \frac{C^d}{C^s} + \epsilon \right]^{-1} \tag{11}$$

3.2.3. The network

There are several possibilities to model the network power consumption ([20],[21]). In this study we make an abstraction of the specific layout of the network. We denote the network power consumption per user as P^n. P^n represents the fully allocated power consumption of the following devices:

- The LAN switch in the client network
- The gateway in the client network
- The wide area network devices (routers and traffic aggregators)
- The gateway in the data center
- The LAN switches in the data center

P^n is determined by the created traffic load on the network. This load is the bandwidth b caused by the thin client protocol. We make again a first order approximation of the power consumption for the model. Based on the results presented in [23] and [24] this is a good approximation for network elements.

$$P^n = P_0^n + \alpha_T^n b \tag{12}$$

3.2.4. Cooling

Due to the concentration of heat-dissipating equipment, considerable efforts are needed to cool data centers. This cooling infrastructure of course also consumes electrical power. Therefore not all electrical power consumed by the datacenter is used for the ICT equipment. This factor is denoted by the Power Usage Effectiveness (*PUE*) [25]:

$$PUE = \frac{P_{tot}^{dc}}{P_{ICT}^{dc}} \tag{13}$$

Since our model should cover multiple cases we will consider the *PUE* accounted for in the power consumption parameters.

3.2.5. Total

The total power of the setup, assuming N^u users is:

$$N^u P^c + N^u P^n + \frac{N^u}{N} P^s \tag{14}$$

In order to compare the power consumption with the desktop scenario we need to divide this by N^u. The power consumed for one thin client is:

$$P^{tc} = P^c + P^n + \frac{1}{N} P^s \tag{15}$$

Substitution in this formula leads to:

$$\begin{aligned} P^{tc} = {} & P_0^c + P_0^n + \left(P_0^s + \alpha_{CPU}^s \lambda_{CPU}^s\right) \frac{1}{N} \\ & + \left(\alpha_T^n + \alpha_{NIC}^s\right) b \end{aligned} \tag{16}$$

DOI: 10.1002/ett

4. Passive State Analysis

In the previous section, it was assumed that all users were active. In this section we will study potential benefits and drawbacks arising from passive users. We refer to this analysis as the *passive state analysis*.

Two mechanisms contribute to reduced power consumption in the client scenario:

- The thin client terminal consumes less power when off-line reducing power consumption in the passive state at the client side.
- Servers can be put in a sleep mode when a number of users go to the passive state, thereby reducing the power consumption in the data center.

The total number of users is still denoted by N^u. The quantities N^u_{act}, N^u_{off} and N^u_{sb} denote the number of clients in the active state, the off state and the standby state respectively. Power consumed by device '*' in these states is represented by P^*_{act}, P^*_{off} and P^*_{sb} respectively. Clearly, we have at any given moment (N^u constant)

$$N^u = N^u_{act} + N^u_{off} + N^u_{sb} \tag{17}$$

as well as the average (averaging per user) power consumption of device '*' ($* = d, c$)

$$P^*_{avg} = \frac{N^u_{act}}{N^u} P^*_{act} + \frac{N^u_{off}}{N^u} P^*_{off} + \frac{N^u_{sb}}{N^u} P^*_{sb} \tag{18}$$

Applying the approximations used in the previous sections for the active state power consumption, we find for the desktop and the thin client terminal:

$$P^d_{avg} = P^d_{0,avg} + \frac{N^u_{act}}{N^u} \alpha^d_{CPU} \lambda^d_{CPU} \tag{19}$$

$$P^c_{avg} = P^c_{0,avg} \tag{20}$$

Similary we get for the server

$$P^s_{avg} = \frac{N^s_{act}}{N^s} P^s_{act} + \frac{N^s_{off}}{N^s} P^s_{off} + \frac{N^s_{sb}}{N^s} P^s_{sb} \tag{21}$$

If we denote the server CPU load in active state as $\lambda^s_{CPU,AS}$ given by equation (10) and the share ratio $N_{act} \triangleq N^u_{act}/N^s_{act}$ we get:

$$\lambda^s_{CPU} = N_{act} \left[\lambda^d_{CPU} \frac{C^d}{C^s} + \epsilon \right] \tag{22}$$

$$\Rightarrow \lambda^s_{CPU} = \frac{N_{act}}{N} \lambda^s_{CPU,AS} \tag{23}$$

This means:

$$P^s_{avg} = P^s_{0,avg} + \frac{N^u_{act}}{N^s} \frac{1}{N} \alpha^s_{CPU} \lambda^s_{CPU,AS} + \alpha^s_{NIC} \frac{N^u_{act}}{N^s} b \tag{24}$$

In the network we also consider three states. In the active state the equipment is performing its full functionality. In the off state the equipment is switched off. In the standby state the equipment has a reduced functionality. Typically network devices in standby operate at reduced power with a bitrate of $128\ \mathrm{kbit/s}$. Additionally they have a small wake up time so the network functionality is not compromised.

At the user premises one can afford to switch off the network equipment. Deeper in the network this is however not possible. The user is not present to activate the equipment and moreover the equipment is shared between multiple users. Therefore defining the network state as switched off or standby is not as straightforward as with the desktops, client terminals and servers. In order to maintain the generality of the model, for networks we will use the term *reduced power states*. These reduced power states will only affect P^n_0 since the bandwidth b is only originating from the active connections. Thus P^n in the passive state becomes:

$$P^n_{avg} = \frac{N^u_{act}}{N^u} P^n_0 + \frac{N^u_{sb}}{N^u} \sum_i f^{n,sb}_i P^{n,sb}_{red,i} + \frac{N^u_{off}}{N^u} \sum_i f^{n,off}_i P^{n,off}_{red,i} + \alpha^n_T b \tag{25}$$

Where $f^{n,*}_*$ denotes the fraction of the representation of a certain reduced power state $P^{n,*}_{red,*}$ for the client terminal in standby or switched off.

4.1. Desktop Scenario

When comparing the power consumption in the passive state (19) to the active state (5) it is obvious that the power consumption of the desktops is reduced by the power saving of the machines that are shut down or in standby and the CPU load not consumed by these machines.

4.2. Thin Client Scenario

For the thin client scenario we have yet to define the state distribution N^s_* of the servers and the passive state of the

DOI: 10.1002/ett

network. Firstly we will consider three scenarios with a fully active network (i.e. no reduced power states). In the first scenario unused servers are not put in standby mode (or even shut down). In the second scenario we will assume that servers can be put in a power saving mode. In the third scenario we will shut down the servers instead of putting them in power saving mode. It is obvious that the second and the third scenario will imply power savings in the model. There are however some drawbacks to these scenarios:

- Reducing the number of active servers while sessions are running requires a flexible migration of these sessions in order not to affect the active users.
- Shutting down servers is less flexible than putting them in standby. The responsiveness of the server management under varying activity of the users will have to be evaluated.

Secondly we will introduce the additional power saving we can get from reduced power states in the network.

4.2.1. *Scenario I: All servers remain active*

When all servers remain active $N^s_{act} = N^s$. This means:

$$N_{act} \triangleq \frac{N^u_{act}}{N^s_{act}} = \frac{N^u_{act}}{N^u} N \tag{26}$$

Using these values we can calculate the power consumption in this scenario. We denote this power consumption as P^{tc}_I.

$$\begin{aligned} P^{tc}_I = P^{tc} - & \\ & \left(\frac{N^u_{off}}{N^u} \left(P^c_{act} - P^c_{off} \right) + \frac{N^u_{sb}}{N^u} \left(P^c_{act} - P^c_{sb} \right) \right) \\ & - \frac{N^u_{sb} + N^u_{off}}{N^u} \\ & \quad \left(\alpha^s_{CPU} \lambda^s_{CPU,AS} \frac{1}{N} + \left(\alpha^n_T + \alpha^s_{NIC} \right) b \right) \end{aligned} \tag{27}$$

P^{tc} is (16) with P^c_0 and P^s_0 given by $P^c_{0,act}$ and $P^s_{0,act}$ respectively.

The power consumption is reduced by two factors. Firstly we see the obvious power saving caused by the clients being shut down or in standby. Secondly the power consumption is further reduced by the lower CPU load and bandwidth consumption.

4.2.2. *Scenario II: Servers in power saving mode when possible*

In this scenario, unused servers are put in a power saving mode. Note that in order to apply this scenario technology to migrate running jobs between the servers that will be put in power saving mode and the servers that remain online is required.

Since only a fraction of $\frac{N^u_{act}}{N^u}$ of users consumes computing cycles, we assume that only this fraction of servers is active or $N^s_{act} = \frac{N^u_{act}}{N^u} N^s$. This translates into:

$$N_{act} \triangleq \frac{N^u_{act}}{N^s_{act}} = N \tag{28}$$

Similarly as in section 4.2.1 we denote this power consumption as P^{tc}_{II}:

$$P^{tc}_{II} = P^{tc}_I - \frac{N^u_{sb} + N^u_{off}}{N^u} \frac{1}{N} \left(P^s_{0,act} - P^s_{0,sb} \right) \tag{29}$$

One sees that the power consumption is further reduced by the power saving of the servers in standby.

4.2.3. *Scenario III: Servers shut down when possible*

In this scenario we use the same assumptions as in the previous section. Only now we shut down the servers instead of putting them in power saving mode or $N^s_{act} = \frac{N^u_{act}}{N}$. This translates into:

$$N_{act} \triangleq \frac{N^u_{act}}{N^s_{act}} = N \tag{30}$$

We get:

$$P^{tc}_{III} = P^{tc}_{II} - \frac{N^u_{sb} + N^u_{off}}{N^u} \frac{1}{N} \left(P^s_{0,sb} - P^s_{0,off} \right) \tag{31}$$

A similar power saving as in the previous scenario can be found.

We can simplify these results by assuming that we physically cut off the power of shut down equipment. That means $P^*_{off} = 0$. Further we assume that the inactive thin clients and desktops are shut down so $N^u_{sb} = 0$. Note that these assumptions mean that we use both the desktop and the thin client solutions in their most energy efficient way.

$$\begin{aligned} P^{tc}_{III} = P^{tc} - \frac{N^u_{off}}{N^u} \Big[& P^c_{act} \\ & + \left(P^s_{0,act} + \alpha^s_{CPU} \lambda^s_{CPU,AS} \right) \frac{1}{N} \\ & + \left(\alpha^n_T + \alpha^s_{NIC} \right) b \Big] \end{aligned} \tag{32}$$

DOI: 10.1002/ett

$$P_{III}^{tc} = \frac{N_{act}^u}{N^u} P^{tc} + \frac{N_{off}^u}{N^u} P_0^n \qquad (33)$$

The power consumption of the thin client solution scales with the number of active users except for the basic network power consumption.

4.2.4. Scenario IV: Reduced Power states in the network

To further scale down power consumption the only remaining option is introducing reduced power states in the network. For the passive network connections we assume one reduced power state P_{red}^n. Using equation (25) we get:

$$P_{avg}^n = \frac{N_{act}^u}{N^u} P_0^n + \frac{N_{off}^u}{N^u} P_{red}^n + \alpha_T^n b \qquad (34)$$

This leads to

$$P_{IV}^{tc} = \frac{N_{act}^u}{N^u} P^{tc} + \frac{N_{off}^u}{N^u} P_{red}^n \qquad (35)$$

$$= P_{III}^{tc} - \frac{N_{off}^u}{N^u} \left(P_0^n - P_{red}^n\right) \qquad (36)$$

It is clear that in order to achieve maximal energy effciency the P_{red}^n needs to be minimal. When $P_{red}^n = 0$ the energy consumption of the thin client solution scales with the number of active users. Note however that this case is only theoretical since we need to maintain a minimal connectivity in the network.

5. Experimental Results

When evaluating practical implementations it is important to gain insight in the power saved by implementing thin clients. Therefore we will evaluate two parameters. We define the saved power as $\Delta P = P^d - P^{tc}$ which will express the power saving for a single user. The second parameter is the power ratio $R = \frac{Pd}{Ptc}$ which expresses the relative power saving between both scenarios. The criterium for power efficiency is:

$$\Delta P > 0 \qquad (37)$$

or stated otherwise:

$$R > 100\% \qquad (38)$$

We evaluate these parameters in function of the server share ratio N and the network power consumption P_0^n. Since we are focussing on standard office applications such as text editors and spreadsheets, we will assume an average load λ_{CPU}^d of 20% which is largely sufficient. For these applications network connectivity is not required so we consider no network power consumption in the desktop scenario. In the thin client scenario the standalone applications run remotely on servers in the data center. The consumed bandwidth will vary between $0\ \mathrm{Mbit/s}$ and $5\ \mathrm{Mbit/s}$ [14]. The server overhead ϵ is considered to be small ($\epsilon \approx 0$). The servers are located in a data center for which we assume a typical *PUE* of 2.

Desktop PC (AMD Athlon 64 3500+™)		Laptop PC (Pentium M™ 2 GHz)	
P_0^d	82.6 W	P_0^d	28.6 W
α_{CPU}^d	13.9 W	α_{CPU}^d	10 W
C^d	1401	C^d	1541

Client Terminal (JackPC™)		Server (AMD Opteron 2212™)	
P_0^c	4 W	P_0^s	217 W
		$P_{0,sb}^s$	15.8 W
		α_{CPU}^s	10.42 W
		α_{NIC}^s	$0.93\ \frac{\mathrm{mW}}{\mathrm{Mbit/s}}$
		C^s	4×1435
		PUE	2

Table 1. Equipment Parameters.

We have measured specific devices which are considered representative for the case. We measured the power consumption of a desktop (AMD Athlon 64 3500+™), a laptop (Pentium M™ 2 GHz), a server (AMD Opteron 2212™) and a thin client device (JackPC™). We have applied different processor loads on these devices. With a feedback loop we introduced short sleep times (approximately 10 ms of sleep) in a running program so we could achieve the requested server load. We applied a linearization at the expected processor load. This was $\lambda = 10\%$ for the desktop and laptop and $\lambda = 100\%$ for the server. For the client terminal there was no dependency on the processor load. The variance on the measurements was $\sigma = 0.2\%$ for the server, $\sigma = 10\%$ for the laptop and $\sigma = 2.7\%$ for the server.

The measured parameters are summarized in table 1. The profile of the power consumption of the server corresponds with the typical behaviour as can be seen in [26]. The bandwidth factor α_{NIC}^s appears to be small compared to the relevant bandwith and the other parameters. We assume the same order of magnitude for α_T^n Therefore we will ignore the factor $(\alpha_T^n + \alpha_{NIC}^s)\, b$.

In order to be able to comply with the QoS requirements for thin clients (as described in section 2) we assume the data center to be located in the access network.

Euro. Trans. Telecomms. **00**: 1–13 (0000)
DOI: 10.1002/ett

	User Prem. Eq.	Access Netw. Eq.	Total
Active state			
ADSL2	1.5 W	1.2 W	2.7 W
VDSL2	6.0 W	1.6 W	7.6 W
PON	12.0 W	0.2 W	12.2 W
Reduced Power State			
ADSL2	0.0 W	0.8 W	0.8 W
VDSL2	0.3 W	1.0 W	1.3 W
PON	0.3 W	0.2 W	0.5 W

Table 2. Power Consumption per User of Network Equipment [27].

[27] mentions target values for the power consumption of the network equipment. We consider three network technologies: ADSL2, VDSL2 and PON. For the access network power consumption of the PON we assume a typical value of $0.2\ \mathrm{W/subscriber}$. For the reduced power state we assume the equipment at the user premises to be switched off and the access network equipment in standby state. The network power consumption values are summarized in table 2.

The power consumption share of network equipment deeper in the network is not accounted for in this case. First of all because, as stated before the servers cannot be too deep in the network. Moreover, that equipment will be shared over a large number of users so the power consumption per subscriber becomes negligible. Note as well that this is why the PON case is the least power efficient. The bandwidth provided by this solution is significantly larger than for the other solutions whereas this is not required. It would be more efficient to share these high bandwidths over a larger number of users and then implement the final connection with an ADSL2/VDSL2 line. This case however is covered by the first two solutions.

We analyse the active state and the passive state. To limit the complexity we will assume that the desktops or client terminals are either active or switched off ($N^u_{sb} = 0$).

5.1. *Active State Analysis*

Figure 2 displays a breakdown in the power consumption for a Desktop PC, a Laptop PC and a Thin Client Setup. Note however that manufacturers limit the power consumption of a laptop as much as possible. When comparing P^d_0 one can see that a laptop PC is roughly three times more efficient than a desktop PC while the two machines have the same functionality. Servers on the other hand are not designed for energy efficiency in the same degree as laptop PCs. When we compare the thin client paradigm in which the main power consumer (the server) is not optimized for its power use to a laptop which is completely optimized for its power use, this is not a fair comparison. Therefore we will focus on comparing the desktop PC with the thin client solution. However, it is to be expected that in the future servers will become a lot more power efficient and data centers will be designed with a lower *PUE*. Then the comparison with laptop PCs can be made.

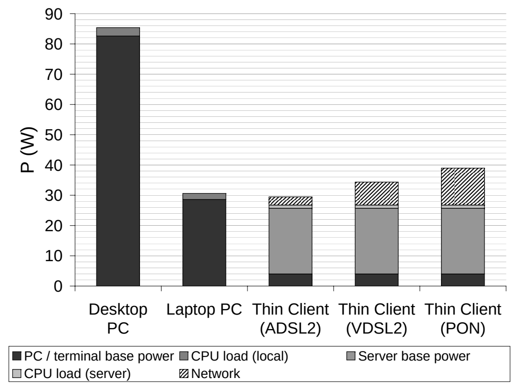


Figure 2. Power consumption of Desktop PC, Laptop PC and Thin Client in active state for office applications($\lambda^d_{CPU} = 20\%$) and N=20.

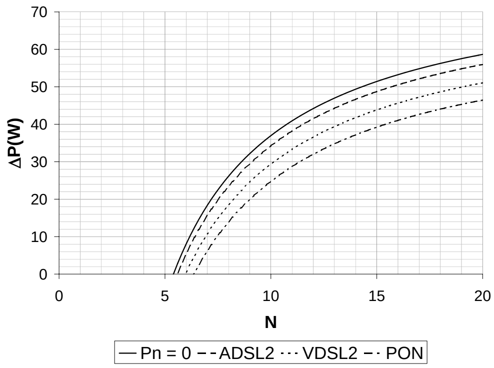
In Figure 2 we can see that compared to the Desktop PC the power consumption of Thin Client Setup is significantly lower. Using ADSL2, the advantage is the largest and when using VDSL2 or PON the power saving decreases slightly.

Figure 3 displays ΔP and R in function of the server share ratio N for the different technologies. Next to the three network technologies the figures include the theoretical case of $P^n = 0$ as well. The power saving is highly dependent on the share ratio on the servers. A minimal share ratio of $N > 5$ in order to be more efficient than the desktop scenario. At the maximal share ratio of $N \approx 20$ ($\lambda^s_{CPU} \approx 1$) power savings up to 50 W (300%) are possible. It is also clear that the choice of network technology can have an important impact on the power saving possibilities. However, the impact of the server power consumption stil remains the most significant.

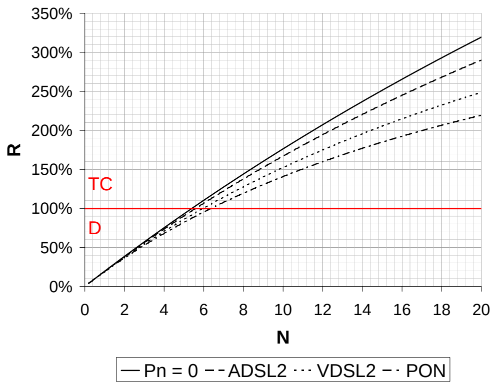
5.2. *Passive State Analysis*

When regarding the passive state analysis the three relevant scenarios are displayed in Figure 4. We are still only

Euro. Trans. Telecomms. **00**: 1–13 (0000)
DOI: 10.1002/ett



(a) ΔP



(b) R

Figure 3. Power saving of Thin Client(TC) scenario towards Desktop(D) scenario in function of Server Share Ratio and Network Power Consumption for office applications($\lambda^d_{CPU} = 20\%$).

considering standard office applications so for λ^d_{CPU} we assumed a value of 20%. We assumed a share ratio of $N = 20$. However, the conclusions are qualitatively similar for different values for the share ratio N.

When $R = 100\%$ the desktop scenario (D) is exactly as power efficient as the thin client scenario (TC). We denote this passive user fraction as $\left(\frac{N^u_{off}}{N^u}\right)_{R=100\%}$.

When all the servers remain active (I) the efficiency degrades approximately linearly in function of the fraction of passive users $\frac{N^u_{off}}{N^u}$. We see that, depending on the network technology, $\left(\frac{N^u_{off}}{N^u}\right)_{R=100\%}$ is approximatelay 60% to 70%. Putting servers in standby (II) or switching them off (III) can lead to large increases in the efficiency: $\left(\frac{N^u_{off}}{N^u}\right)_{R=100\%} = 77\% - 95\%$. If the network power consumption is low (ADSL2) the increase is more significant than when the network power consumption is higher (PON). Introducing reduced power states in the network (IV) further increases the energy efficiency: $\left(\frac{N^u_{off}}{N^u}\right)_{R=100\%} = 97\% - 98\%$. These improvements are more significant with larger differences between P^n_0 and P^n_{red}.

Between ADSL2, VDSL2 and PON there is a trade off. When all users are active ADSL2 is clearly more advantageous. However, for $N^u_{off}/N_u > 97\%$ PON is more efficient due to the large gap between the P^n_0 and P^n_{red}. This is displayed in Figure 5. This trade off will be important for implementations where large passive user fractions during long periods of time can be expected, in this case for $\frac{N^u_{off}}{N^u} > 97\%$.

The passive state analysis clearly shows that the choice of a low power network technology with the possibility of reduced power states is required in order to assure power efficiency even with a large number of passive users.

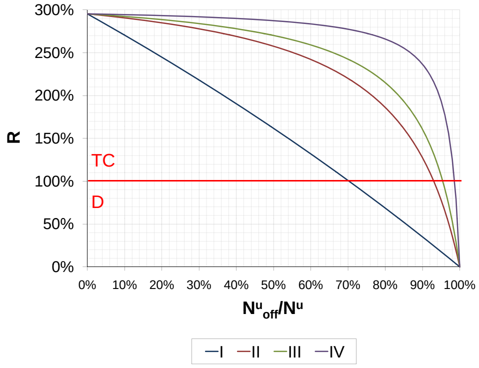
6. Conclusions

ICT represents a relevant fraction of the worldwide energy production. The growth rate of this fraction is difficult to sustain. We created an analytical model in order to determine if the thin client paradigm is more power efficient than the desktop PC. Using experimental data different specific cases were reviewed.

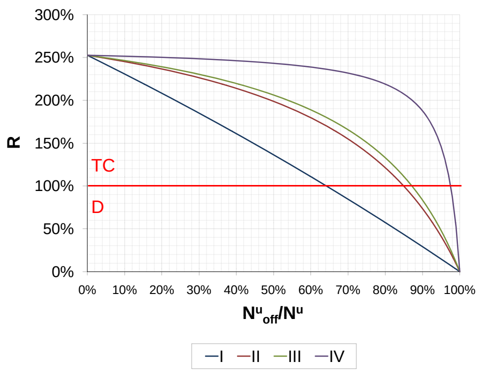
These cases displayed that power savings up to $2/3$ of the desktop power consumption can be saved by replacing with a thin client setup. In [16] a power saving of approximately 50% is achieved for both the entire life cycle and the operation phase. Considering the use of different hardware, these results are consistent. [21] achieves a power saving of 83%. Note that this paper is focussing on a different kind of virtualization.

The power saving potential is impaired by a reduced efficiency when a fraction of the users is passive. This can be mitigated by selectively switching off servers when reduced activity occurs. Secondly, introducing reduced power states in the network make the thin client paradigm more power efficient for idle user ratios up to 95%.

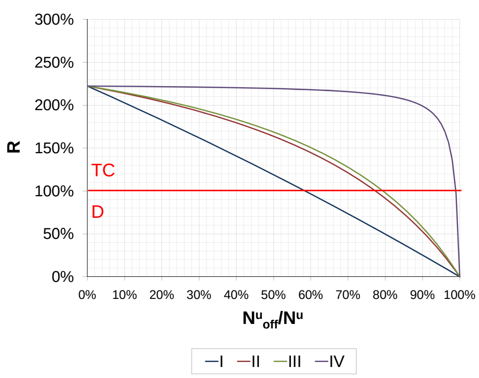
Further optimizations of the power consumption of the servers in the data centers will be required. Currently the efficiency of laptop PCs is approximately the same as the thin client scenario. The power efficiency of servers will have to be brought on the same level as laptop PCs and the data centers Power Usage Effectiveness (*PUE*) will have to



(a) ADSL2



(b) VDSL2



(c) PON

Figure 4. Power Saving Ratio of Thin Client(TC) scenario towards Desktop(D) scenario in function of the fraction of passive users for office applications($\lambda^d_{CPU} = 20\%$) for each of the relevant scenarios (I - IV) in the passive state analysis.

be improved in order to achieve the potential described in this work.

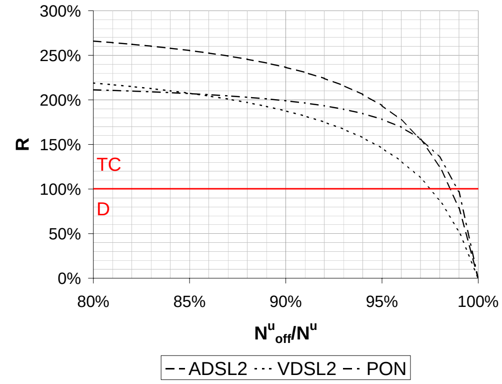


Figure 5. Power Saving Ratio of Thin Client(TC) scenario towards Desktop(D) scenario using Reduced Network Power States for office applications($\lambda^d_{CPU} = 20\%$).

ACKNOWLEDGEMENTS

The work described in this paper was carried out with the support of the BONE project ("Building the Future Optical Network in Europe"), a Network of Excellence funded by the European Community's Seventh Framework. Part of the research leading to these results was done for the MobiThin Project and has received funding from the European Community's Seventh Framework (FP7/2007-2013) under grant agreement nr 216946.

References

1. United Nations. Millenium Development Goals Indicators: Internet Users per 100 population. http://unstats.un.org/unsd/mdg/SeriesDetail.aspx?srid=605.
2. Gantz JF, Chute C, Manfrediz A, Minton S, Reinsel D, Schlichting W, Toncheva A. The diverse and exploding digital universe, an updated forecast of worldwide information growth through 2011. *IDC White Paper* March 2008; .
3. Koomey J. Estimating total power consumption by servers in the U.S. and the world February 2007.
4. Cole D. *Computers and the Environment: Understanding and Managing their Impacts (Eco-Efficiency in Industry and Science)*, chap. Chapter 7, Computers and the environment. Springer-Verlag New York, Inc.: Secaucus, NJ, USA, 2003; 136–138.
5. Pickavet M, Vereecken W, Demeyer S, Audenaert P, Vermeulen B, Develder C, Colle D, Dhoedt B, Demeester P. Worldwide energy needs for ICT: the rise of power-aware networking. *IEEE ANTS 2008*, 2008.
6. Kawadia V, Kumar P. Principles and protocols for power control in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications* January 2005; **23**(1):76–88.
7. Gupta M, Singh S. Greening of the Internet. *SIGCOMM'03*, 2003.
8. Hwang CH, Wu ACH. A predictive system shutdown method for energy saving of event-driven computation. *ACM Trans. Des. Autom. Electron. Syst.* 2000; **5**(2):226–241.
9. Chung EY, Benini L, Bogliolo A, Lu YH, De Micheli G. Dynamic power management for nonstationary service requests. *Computers, IEEE Transactions on* Nov 2002; **51**(11):1345–1361.
10. Orgerie AC, Lefevre L, Gelas JP. Save watts in your grid: Green strategies for energy-aware framework in large scale distributed

systems. *Parallel and Distributed Systems, 2008. ICPADS '08. 14th IEEE International Conference on*, 2008; 171–178.

11. IEEE 802.3 Study Group on Energy-Efficient Ethernet. Http://www.ethernetalliance.org.
12. The green grid. Http://www.thegreengrid.org.
13. Lai A, Nieh J. On the performance of wide-area thin-client computing. *ACM Trans. Comput. Syst.* 2006; **24**(2):175–209.
14. Deboosere L, De Wachter J, De Turck F, Dhoedt B, Demeester P. Thin Client Computing Solutions in Low- and High-Motion Scenarios. *International Conference on Networking and Services (ICNS)*, 2007; 38.
15. Simoens P, Praet P, Vankeirsbilck B, De Wachter J, Deboosere L, De Turck F, Dhoedt B, Demeester P. Design and implementation of a hybrid remote display protocol to optimize multimedia experience on thin client devices. *2008 Australasian Telecommunications Networks and Applications Conference*, Adelaide, Australia, 2008.
16. Knermann C, Hiebel M, Pflaum H, Rettweiler M, Schroder A. Environmental comparison of the relevance of pc and thin client desktop equipment for the climate. *Fraunhofer UMSICHT* 2008; .
17. Rivoire S, Ranganathan P, Kozyrakis C. A comparison of high-level full-system power models. *HotPower*, Zhao F (ed.), USENIX Association, 2008.
18. Fan X, Weber WD, Barroso LA. Power provisioning for a warehouse-sized computer. *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*, ACM Press: New York, NY, USA, 2007; 13–23.
19. Heath T, Diniz B, Carrera EV, Jr WM, Bianchini R. Energy conservation in heterogeneous server clusters. *PPoPP '05: Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming*, 2005; 186–195.
20. Baliga J, Ayre R, Sorin W, Hinton K, Tucker R. Energy consumption in access networks. *Optical Fiber communication/National Fiber Optic Engineers Conference, 2008. OFC/NFOEC 2008. Conference on*, 2008; 1–3.
21. Hlavacs H, Weidlich R, Hummel KA, Houyou AM, Berl A, de Meer H. Distributed energy efficiency in future home environments. *Annales des Télécommunications* 2008; **63**(9-10):473–485.
22. SPEC. Http://www.spec.org.
23. Wang HS, Peh LS, Malik S. A power model for routers: Modeling alpha 21364 and infiniband routers. *High-Performance Interconnects, Symposium on* 2002; :21.
24. Ye TT, Micheli GD, Benini L. Analysis of power consumption on switch fabrics in network routers. *DAC '02: Proceedings of the 39th conference on Design automation*, ACM, 2002; 524–529.
25. Green grid metrics: Describing datacenter power efficiency 2007.
26. Orgerie AC, Lefevre L, Gelas JP. How an experimental grid is used: The grid5000 case and its impact on energy usage. *Proc. 8th IEEE Int. Symp. on Cluster Computing and the Grid (CCGrid2008)*, 2008.
27. European Commission. Code of conduct on energy consumption of broadband equipment version 2 July 2007.

AUTHORS' BIOGRAPHIES

WILLEM VEREECKEN received his M. Sc. degree in Electrical Engineering from Ghent University, Belgium, in July 2005. He is a Ph.D. student affiliated with the Department of Information Technology of Ghent University. His research is funded by a Dehousse Ph.D. grant from Ghent University. His main research interest is the design of energy efficient network architectures.

LIEN DEBOOSERE received her M. Sc. degree in Computer Science Engineering from Ghent University, Belgium, in July 2005. She is a Ph.D. student affiliated with the Department of Information Technology of Ghent University. Her research is funded by a Ph.D. grant from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT Vlaanderen). Her main research interest is the design of architectures for wide area mobile thin client computing.

PIETER SIMOENS received his M. Sc. degree in Electronic Engineering from Ghent University, Belgium, in July 2005. Since October 2005, he is affiliated as research assistant with the Department of Information Technology of Ghent University. He is funded through the Fund for Scientific-Research-Flanders, Belgium (F.W.O.-V.). His main research interest is the development of intelligent and adaptive network architectures and protocols designed for thin client computing and autonomous environments. He cooperated in the IST FP6 project MUSE and is now work package leader for the protocol design in the IST FP7 project MobiThin.

BRECHT VERMEULEN received his Electronic Engineering degree in 1999 from the Ghent University, Belgium. In June 2004, he received his PhD degree for the work entitled 'Management architecture to support quality of service in the internet based on IntServ and DiffServ domains.' at the department of Information Technology of Ghent University. Since June 2004, he is leading a research team within the IBCN group of Prof. Demeester which investigates network and server performance and quality of experience in the fields of video, audio/voice and multiple play. Since the start of IBBT (Interdisciplinary institute for BroadBand Technology) in 2004, he leads also the IBBT Technical Test centre iLab.t in Gent, Belgium.

DIDIER COLLE received a M. Sc. degree in electrotechnical engineering from the Ghent University in 1997. Since then, he has been working at the same university as researcher in the department of Information Technology (INTEC). He is part of the research group INTEC Broadband Communication Networks (IBCN) headed by prof: Piet Demeester. His research lead to a Ph.D degree in February 2002. He was granted a postdoctoral scholarship from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) in the period 2003-2004. His research deals with design and planning of communication networks. This work is focussing on optical transport networks, to support the next-generation Internet. Up till now, he has actively been involved in several IST projects (LION, OPTIMIST, DAVID, STOLAS, NOBEL and LASAGNE), in the COST-action 266 and 291 and in the ITEA/IWT TBONES project. His work has been published in more than 100 scientific publications in international conferences and journals.

CHRIS DEVELDER received the M.Sc. degree in computer science engineering and a Ph.D. in electrical engineering from Ghent University (Ghent, Belgium), in July 1999 and December 2003 respectively. From October 1999 on, he has been working in the Department of Information Technology (INTEC), at the same university, as a Researcher for the Research Foundation-Flanders (FWO), in the field of network design and planning, mainly focusing on optical packet switched networks. In January 2004, he left University to join OPNET Technologies, working on transport network design and planning. In September 2005, he re-joined INTEC at Ghent University as a post-doctoral researcher, and as a post-doctoral fellow of the FWO since October 2006. Since October 2007 he holds a part-time professor position at Ghent University. He was and is involved in multiple national and European research projects (IST Lion, IST David, IST Stolas, IST Phosphorus, IST E-Photon One, BONE). His current research focuses on dimensioning, modeling and optimizing optical Grid networks and their control and management, as well as multimedia and home network software and technologies. He is an author or co-author of over 50 international publications.

MARIO PICKAVET received an M.Sc. and Ph.D. degree in electrical engineering, specialized in telecommunications, from Ghent University in 1996 and 1999, respectively. His current research interests are related to broadband communication networks (WDM, IP, (G-)MPLS, Ethernet, OPS, OBS) and include design, long-term planning, techno-economical analysis and energy efficiency of core and access networks. Special attention goes to Operations Research techniques that can be applied for routing and network design. In this context, he is currently involved in several European and national projects, such as the Network of Excellence "Building the Future Optical Network in Europe" (BONE). He has published about 200 international publications, both in journals (IEEE JSAC, IEEE Comm. Mag., Journal of Lightwave Technology, Eur. Trans. on Telecommunications, Photonic Network Communication, ...) and in proceedings of conferences. He is one of the authors of the book 'Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS'.

BART DHOEDT received a degree in Engineering from Ghent University in 1990. In September 1990, he joined the Department of Information Technology, Ghent University. His research, addressing the use of micro-optics to realize parallel free space optical interconnects, resulted in a Ph.D. degree in 1995. After a 2 year post-doc in opto-electronics, he became professor at the Department of Information Technology. He is responsible for courses on algorithms, programming and software development. His research interests are software engineering and mobile & wireless communications. His current research addresses software technologies for communication networks, peer-to-peer networks, mobile networks and active networks.

PIET DEMEESTER received the Masters degree in Electro-technical engineering and the Ph.D degree from Ghent University, in 1984 and 1988, respectively. In 1992 he started a new research activity on broadband communication networks resulting in the IBCN-group (INTEC Broadband communications network research group). Since 1993 he became professor at Ghent University where he is responsible for the research and education on communication networks. The research activities cover various communication networks, including network planning, network and service management, telecom software, internetworking, network protocols for

QoS support, etc. He is member of the editorial board of several international journals and has been member of several technical program committees (ECOC, OFC, DRCN, ICCCN, IZS).

DOI: 10.1002/ett